

# Distributed Co-phasing for Binary Consensus in Wireless Sensor Networks

Harish Vangala and Chandra R. Murthy, *Senior Member, IEEE*

**Abstract**—We address the problem of binary consensus over noisy, fading channels. We propose a new transmission model using Distributed Co-phasing that ensures the system to reach accurate consensus quite efficiently. We propose a simple linear update rule at nodes based on LMMSE and see that its performance is close to the ML estimate. Then we formulate the problem as a Markov chain and find an approximation for the second eigenvalue of the transition probability matrix which characterizes its transient behavior. We present simulation results to validate the performance.

**Index Terms**—Distributed cophasing, DCP, wireless sensor networks, binary consensus, detection consensus.

## I. INTRODUCTION

The problem of *consensus* refers to any problem where a number of individual elements are expected to agree upon a common value to a variable. Such a problem has been well-known and is quite fundamental to a variety of applications like cooperative control, surveillance and security, smart homes, target tracking systems [1], [2]. Initially, the problem was studied extensively by control and systems community in the past decade in distinct setups. In their models, a number of elements called as *agents*, attempt to come to a consensus upon a value which is usually a continuous value. Vehicle formations, rendezvous problem and coordinated decision making etc are few examples for this problem [1].

In the respective models, the agents are viewed as the vertices of a graph and need not have communication-link with all other nodes. Hence, every agent communicates its value (possibly a function) to its neighbors and a similar operation is followed by all nodes. A periodic updating of nodes upon receiving values from its neighbors is expected to achieve consensus asymptotically and convergence is studied rigorously [1]. Recently the problem has gained considerable interest in communication and signal processing community and a class of algorithms called gossip algorithms have been a quite active area of research [3]–[9]. However, the communication links are mostly assumed to be fairly good or noiseless. A relatively small focus was made on the problem with link-uncertainties till recent time [2].

When the data to achieve consensus upon takes values from a finite set, the problem is called as *detection consensus* and such a problem is attached to the physical layer of the system. In this model, link uncertainties play a major role to characterize the performance of the system. Further, it makes

the system more challenging to analyze. There has been a small focus towards this problem and to the best of author's knowledge, the main contributions are due to [2], [10]. This work on detection consensus with noise and fading, considered two very different cases of transmission models being called as *fusion* and *diversity* models. However, a guaranteed behavior of convergence is still an open problem. Though the diversity scheme is said to achieve accurate consensus asymptotically, it is seen to be much slow in its performance. This work does the very first analysis of a practical model for detection-consensus and provides a foundational basis for the analysis of a class of potentially much complex consensus systems.

In a common wireless network system, pilot assistance is used in communication to obtain appropriate channel state information (CSI) at the ends, so that nodes utilize the CSI to achieve an efficient communication in the system. It is known that a typical wireless sensor network contains nodes that are inexpensive, battery driven and are with limited computational capabilities. For this reason, efficient distributed algorithms with pilot assistance are desired, to perform operations like water-filling, without having to require expensive RF components for fine power-control [11].

Pilot assisted distributed cophasing is a system proposed to elicit the benefits of distributed beamforming for uplink communication over a wireless sensor network. This system significantly reduces the channel feedback and power control requirements, by utilizing the reciprocity property of the channel [12] and significantly improves the SNR at the receiver without using any power control at transmitters. The system is evaluated as an efficient transmission scheme that provides a competitive performance at a very less complexity, when compared to the alternative schemes existing in the literature [11].

An improved system performance in detection consensus in terms of convergence rate and the probability of achieving accurate consensus is still strongly desired over the proposed systems till date. The current paper addresses this issue and tries to propose a new scheme using *distributed co-phasing* transmission to improve the performance. We try to demonstrate in this paper, an analysis of the proposed scheme and the performance gains achievable and show them via simulations.

The following are the contributions of this paper.

- 1) We propose a new system model for achieving binary consensus in a wireless sensor network.
- 2) We present a simple node update rule based on Least Minimum Mean Square Error (LMMSE) estimation of

the sum of votes at each node. We show via simulations that its performance is close to that of an ML estimator.

- 3) We model the problem as a Markov chain and characterize its transient and steady state behavior by the second eigen value of the average transition probability matrix.
- 4) We find an efficient approximation for the second eigen value of the system, to seek insights into the transient behavior of the system.

The rest of the paper is organized as follows. We first introduce the system model and formulate the problem. We provide two node update rules and with the help of those update rules, we look at the consensus aiming system as a Markov chain and derive conclusions over its transient and steady state behavior. We then derive an approximate expression for the second eigenvalue of the transition probability matrix that gives insights into the convergence rate of the algorithm. In the later sections we give simulation results of the proposed system and provide conclusions at the end.

## II. SYSTEM MODEL AND PROBLEM FORMULATION

### A. System Model

Consider a set of nodes  $\{1, 2, \dots, N\} \triangleq \mathcal{N}$  separated in space and connected wirelessly with a fully-connected graph topology.

At any instance in time, each node in the network carries a binary data value with antipodal representation, where  $-1$  represents binary *zero*. We define a *state* of the network as the collection of all the data values collected in an  $N$ -tuple in the order of the node indices. Therefore, a *state* is an  $N \times 1$  dimensional binary vector denoted by  $\phi \triangleq [\phi_1, \phi_2, \dots, \phi_N]^T$ . The set of all possible states is denoted by  $\Phi \triangleq \{\phi^{(1)}, \phi^{(2)}, \dots, \phi^{(M)}\}$ , with  $M = 2^N$ . We call the state with all entries equal to  $-1$  as the state of *all-zero* and consider it as state  $\phi^{(1)}$ . And the state with all entries equal to  $+1$  is called as *all-one* state and is considered as state  $\phi^{(M)}$ . Further, we define  $\Phi = \Phi_0 \cup \Phi_1$ , where  $\Phi_1$  denotes the set of states where one is in majority or  $\{\phi \in \Phi : \sum_j \phi_j \geq 0\}$  and  $\Phi_0 = \Phi \setminus \Phi_1$ . Note that all conflict cases of majority are decided in favor of  $+1$  and such states are accounted into  $\Phi_1$ .

Time is slotted and we consider an *iteration* as the time duration in number of time slots, to complete two phases of transmissions, namely pilot transmission phase and data transmission phase. The pilot transmission phase is the phase in which each node one after the other broadcasts the pilot symbols so that nodes acquire perfect channel state information (perfect-CSI) from all other nodes. In the data transmission phase, each node acts as a receiver one after the other while rest all nodes transmit data to it. For this transmission, each node modulates its own single-bit data using a unit energy BPSK constellation  $\{\pm 1\}$ . Then, as per the distributed co-phasing transmission, each transmitting node transmits data added with a phase angle equal and opposite to the phase of the known channel state from itself to the receiver node. This effectively causes all the data signals from transmitting

nodes to combine coherently with the same zero phase, at the receiver node, thus maximizing the instantaneous *signal to noise ratio* (SNR) under no power-control mechanism. All the transmissions are assumed to be synchronous.

In practice, though it is challenging to achieve a precise channel phase estimation and carrier synchronization, it is seen that up to even moderately large phase errors due to synchronization and imperfect channel estimation, the performance doesn't get affected significantly [11], [13].

All channels are assumed to be quasi-static block faded where the channel gains are assumed to be constant during a complete *iteration*. Further, channel is assumed to be reciprocal where the channel gains are same irrespective of the direction of transmission between any two nodes. This assumption simply requires that the transmit and receive RF chain components are well calibrated. The channel gain values are assumed to take values according a circular Gaussian distribution with unit variance between all pairs of nodes. The additive noise at a receiver node  $i$  during the data-transmission phase is assumed to take values according to a circular Gaussian distribution with variance  $\sigma_i^2$ .

The channel matrix with elements equal to the channel gain between each pair of nodes at time instant  $t$  is a symmetric  $N \times N$  dimensional square matrix denoted by  $H(t)$ . We define *channel to noise ratio* (CNR) at a receiver as the ratio between the effective channel gain variance to the additive noise variance. Since in data-transmission phase,  $(N - 1)$  independent transmissions are simultaneously active towards the receiver  $i$ , we take the effective CNR at node  $i$  as  $\frac{N-1}{\sigma_i^2}$ .

At the end of an iteration, with the received data and its own data each node estimates what bit the majority nodes take and adjusts itself to it, expecting that once all nodes do the same the system reaches consensus. Hence the network changes its state after each iteration. We denote the state of the network at time  $t$  as  $\mathcal{D}(t)$ , where  $\mathcal{D}(t) \in \Phi$ . Note that under noise and fading  $\{\mathcal{D}(t) : t \in \mathbb{Z}^+\}$  is a random process.

At any point of time, if the network reaches a state where all nodes take the same value then we say that the network is in *consensus*. Further, if each node takes the value equal to the majority bit in the initial state of the network, then we say that the network is in *accurate consensus*. The goal of the system is to reach *accurate consensus* after a finite iterations. But as we see in later sections, at any point of time, there exists a non-zero probability of not reaching the consensus or moving out of consensus due to fading and noise.

The rule for estimating the majority bit at the end of an iteration is a function of the received data in data transmission phase and its own data. We call this as node's *update rule*. At each iteration  $t$  each node  $i$  attains a pair of scalar values namely  $(x_i(t), y_i(t))$ , where  $x_i(t)$  is node's own data and  $y_i(t)$  is:

$$y_i(t) = \sum_{\substack{j=1 \\ j \neq i}}^N |h_{ij}(t)| x_j(t) + n_i(t), \quad (1)$$

where,  $n_i(t) \sim \mathcal{CN}(0, \sigma_i^2)$  and  $h_{ij}(t) \sim \mathcal{CN}(0, 1) \forall i, j \in \mathcal{N}$ .

Under a given update rule, we denote by  $\gamma_{\phi^{(k)}}^{(i)}$  as the probability of a node choosing +1 as its decision. i.e.

$$\gamma_{\phi^{(k)}}^{(i)} = \Pr \{x_i(t+1) = +1 | \mathcal{D}(t) = \phi^{(k)}, H(t)\}$$

Then the probability of system transitioning from state  $\phi^{(m)}$  to  $\phi^{(n)}$  at time instant  $t$  is denoted by  $P_{mn}(t)$ , an element of the set of all transition probabilities, collected in a matrix denoted by  $P(t)$ . We use  $1_{\{\cdot\}}$  to denote an indicator function that takes one when the condition in braces is true and zero other wise.

$$\begin{aligned} P_{mn}(t) &= \Pr \{D(t+1) = \phi^{(n)} | \mathcal{D}(t) = \phi^{(m)}\} \\ &= \prod_{i=1}^M \left\{ 1_{\{\phi_i^{(n)} = +1\}} \gamma_{\phi^{(m)}}^{(i)} + 1_{\{\phi_i^{(n)} = -1\}} (1 - \gamma_{\phi^{(m)}}^{(i)}) \right\} \end{aligned} \quad (2)$$

$$(3)$$

Under the system model described thus far, devising an update rule for nodes and characterizing the resulting system's performance forms the rest of the solution for the problem of consensus addressed in this paper. In the next section, we see two specific update rules namely an ML-rule and an LMMSE-rule.

### III. NODE UPDATE RULES FOR THE CONSENSUS ALGORITHM

Given a pair of scalars  $(y_i, x_i)$ , a node estimates which bit is in majority in the current state. For this we choose an ML update rule that picks up the most probable bit as the node's decision as majority, by evaluating the probability of each state being the current state of the network, given  $(y_i, x_i)$ . Clearly this minimizes the probability of a node choosing a minority as its decision and therefore maximizing the instantaneous probability of the network's state being equal to accurate consensus at the end of the current iteration. Alternatively we propose a simple linear rule on the pair of scalars  $(y_i, x_i)$  to estimate respective bit-sum in the current state based on linear minimum mean square error (LMMSE) estimation and thereby estimating the majority bit by looking at the sign of the estimated sum.

In devising any particular decision rule, it should be noted that a node initially is unaware of the distribution or any specific statistics of another node's data. Such a knowledge would help in estimating the majority more precisely. However, in such a case, the node has to empirically evaluate the desired parameters by decoding the received data and storing it, thereby giving rise to a much challenging system model for the analysis. As a basic model, we assume that the nodes won't know or attempt to empirically calculate any of such statistics. Hence, we consider as per ML rule that the node assumes that each other node takes values from a uniform Bernoulli distribution and proceeds.

#### A. Maximum-Likelihood (ML) update rule

The ML update rule that estimates the most likely majority given the received data, is written as follows:

$$x_i(t+1) = \begin{cases} 1, & \text{if } \Theta^{(i)} \geq 0.5 \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

where,

$$\begin{aligned} \Theta^{(i)} &\triangleq \Pr \{ \mathcal{D}(t) \in \Phi_1 | (x_i(t), y_i(t)), H(t) \} \\ &= \frac{\sum_{\phi \in \Phi_1} \Pr \{ (x_i(t), y_i(t)) | \mathcal{D}(t) = \phi, H(t) \}}{\sum_{\phi \in \Phi} \Pr \{ (x_i(t), y_i(t)) | \mathcal{D}(t) = \phi, H(t) \}} \end{aligned} \quad (5)$$

and,

$$\Pr \{ (x_i(t), y_i(t)) | \mathcal{D}(t) = \phi, H(t) \} = \begin{cases} 0, & \text{if } x_i(t) \neq \phi_i \\ \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{|x_i(t) - \phi_i|^2}{2\sigma_i^2}}, & \text{o.w.} \end{cases} \quad (6)$$

$$\text{where, } \xi = \left( y_i(t) - \sum_{j \neq i} |h_{ji}| \phi_j \right)$$

As a part of ML rule, for the decision at each time instant a node has to compute  $2^N$  number of probability elements using (6) to compute the decision statistic  $\Theta^{(i)}$ .

#### B. LMMSE update rule

Since a sensor node is usually expected to have limited computational capabilities, larger computations at sensor nodes is usually not desirable and decision rules with low complexity are expected. One proposal would be to choose a linear decision rule minimizing the average squared error between the estimated sum of bits and original sum of bits as per Linear Minimum Mean Squared Error (LMMSE) estimation rule. This can be mathematically written for the current problem as follows.

$$\hat{s}'_{i,L} = \arg \min_{\hat{s}'_i} \mathbb{E} [ \|\hat{s}'_i - s'\|_2^2 ], \quad (7)$$

where,

$$\hat{s}'_i = \alpha_i y_i + \beta_i,$$

$$y_i = \sum_{j \neq i} |h_{ji}| x_j + n_j,$$

$$s' = \sum_{j \neq i} x_j ; \quad \alpha_i, \beta_i \in \mathbb{R}.$$

Here,  $\hat{s}'_{i,L}$  denotes the LMMSE estimator of  $\sum_{j \neq i} x_j$  with  $\alpha_i^*, \beta_i^*$  as the desired linear coefficients obtained by solving the optimization problem above, over the variables  $\alpha_i, \beta_i$ . It can be seen that for an unbiased LMMSE estimator with the known channel gain values, the solution is as follows.

$$\alpha_i^* = \frac{\sum_{j \neq i} |h_{ji}|}{\sum_{j \neq i} |h_{ji}|^2 + \sigma_i^2}, \quad \beta_i^* = 0, \quad \forall i \in \mathcal{N}. \quad (8)$$

Then the required sum estimator is given by:

$$\hat{s}_{i,L} = (\hat{s}'_{i,L} + x_i)$$

Now as we have the sum-estimator readily available at the receiver, the decision rule is simply,

$$x_i(t+1) = \begin{cases} 1, & \text{if } \hat{s}_{i,L} \geq 0, \\ 0, & \text{if } \hat{s}_{i,L} < 0. \end{cases} \quad (9)$$

It can be seen that while ML-estimation takes  $\mathcal{O}(2^N)$  computations, LMMSE decision rule takes  $\mathcal{O}(N)$  computations. Further, we see later via simulations that this simple LMMSE based decision rule gives probability of error performance, quite close to that of a much complex ML-decision rule.

#### IV. CONSENSUS ANALYSIS

From the system model it can be inferred that a node takes an update in its own estimated binary data, merely based on the currently received scalar  $y_i$ , the set of channel states in current iteration and its own data in current iteration. Hence, a node's update purely depends on the current state of the system and the current state of the channel. Hence, it is easy to verify that each node's binary data evolves as a Markov chain and hence, the over all system state  $\{\mathcal{D}(t) : t \in \mathbb{Z}^+\}$  evolves as a Markov chain.

Further, as the channel state varies at each iteration randomly, it is a *time variant* Markov chain wherein the transition probability matrix (t.p.m.) of the Markov chain changes with time. We denote this t.p.m by  $P(t)$ , at time instant  $t$ .

To seek insights into the system's transient and steady state performance, we consider two cases where additive noise is present and is absent. In other words, we consider the first case as the system with noise variance equal to zero. In the second case, we consider the system with strictly positive noise variance.

##### A. Zero noise variance

When noise variance becomes zero, with the two decision rules described above, we can observe the following. Recall the notation  $\Phi, \phi^{(1)}, \phi^{(M)}$  representing the set of all possible states, and the specific states of consensus namely *all-zero* and *all-one* respectively.

1) *ML-decision rule*: When additive noise is zero, the ML decision will be able to find the exact transmitted vector just by looking up the scalar value  $y_i$  in all possible signed summation of channel gains given by  $\sum_{j \neq i} |h_{ji}|x_j(t)$ .

Hence, the node by ML-rule will be able to estimate what the majority is, precisely. As a result, every node after the

first iteration itself, will be able to adjust itself to the majority exactly, thereby reaching an accurate consensus in one-step. Then, the effective transition probability matrix at any time  $t$  will have rows equal to point mass p.m.f's where every previous state transits to either  $\phi^{(1)}$  or  $\phi^{(M)}$  with probability one, based on the majority bit being a *zero* or a *one*. In other words,

$$P_{ij}(t) = \Pr \left( \mathcal{D}(t+1) = \phi^{(j)} | \mathcal{D}(t) = \phi^{(i)} \right) \quad (10)$$

$$= \begin{cases} 1, & \text{if } \underline{1}^T \phi^{(i)} \geq 0 \text{ \& } j = M \text{ (or)} \\ & \text{if } \underline{1}^T \phi^{(i)} < 0 \text{ \& } j = 1 \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

It gives rise to a transition probability matrix (TPM) where all columns except first and last are zeros. Such a TPM corresponds to an absorbing Markov chain with absorbing states being the states of *all-zero* and *all-one*. Further, the time to get absorbed is precisely one step.

It is easy to verify that the vectors given by  $[1, 0, \dots, 0]^T$  and  $[0, 0, \dots, 0, 1]^T$  give two left eigenvectors corresponding to the eigenvalue 1. From the characteristic equation of the transition probability matrix  $P(t)$ , it can also be verified that there are exactly two eigenvalues equal to one while rest all eigen values are zero.

2) *LMMSE decision rule*: With the absence of noise, an LMMSE rule will effectively be looking at the sign of the quantity given by  $y_i = (x_i + \alpha_i^* \sum_{j \neq i} |h_{ji}|x_j)$ . Hence larger channel gains dominate and may result in giving an opposite sign to that of the majority bit. When there are distinct bits (distinct signs) in  $x_j$ 's, both the signs are possible for  $y_i$ , depending upon the values of index  $i$  and  $|h_{ji}|$ 's. Further, this implies that given a channel state ( $|h_{ji}|$ 's), every current state  $\phi^{(i)}$  chooses a specific next state with probability one, as each node's decision is a constant under the absence of noise.

However, when the system is in *all-zero* or *all-one* state (states of consensus), the channel gains won't affect the sign of  $y_i$  and hence, every node decides precisely in favor of the majority bit and hence continues to stay in the same state. Further, this is true only for this pair of states, since by choosing appropriate channel gain values, we can construct a range of channel states where the system will change its state when its previous state is a state of non-consensus, and hence the node will definitely come out of that state after a finite number of steps, unlike *all-zero* and *all-one*.

The above two observations lead to confirm that as an average dynamical system, the Markov chain has two absorbing states at any point of time. And any other state cannot be an absorbing state for a long time.

As a result, the transition probability matrix at every point of time will have first and last row being equal to point mass p.m.f's with first row taking state  $\phi^{(1)}$  with probability one and last row taking  $\phi^{(M)}$  with probability one. i.e.  $P_{ii}(t) = 1$  if  $i = 1$  or  $M$ .

From the theory of absorbing Markov chains, we can confirm that with in a finite mean number of steps, the system reaches the state of consensus, with probability one. However, there exists a relatively small probability to go towards the opposite state of accurate consensus.

With the given first and last row structures in the row-stochastic TPM, it is easy to verify that there are at least *two* eigenvalues exactly equal to 1 with corresponding left eigenvectors  $[1, 0, \dots, 0]^T$  and  $[0, 0, \dots, 0, 1]^T$ .

From the two cases discussed above, it can be confirmed that with either of the decision rules, the system is bound to reach the state of consensus with probability one, when noise is absent. Further, the second largest eigen value is exactly equal to 1, in both the cases.

### B. Non-zero noise variance

When noise-variance is non-zero, we can see that given a channel instantiation, the probability of a node updating its own value to '1' with LMMSE decision rule is given by:

$$\begin{aligned}\gamma_\phi^{(i)} &= \Pr\{x_i(t+1) = +1 | \mathcal{D}(t) = \phi, H(t)\} \\ &= Q\left(\frac{-\hat{s}_{i,L}}{\sigma_i \alpha_{i,L}}\right) \\ &= Q\left(\frac{-1}{\sigma_i} \left\{ \frac{1}{\alpha_{i,L}^*} x_i + \sum_{j \neq i} |h_{ji}| x_j \right\}\right) \quad (12) \\ \text{where, } \alpha_{i,L}^* &= \frac{\sum_{j \neq i} |h_{ji}|}{\sum_{j \neq i} |h_{ji}|^2 + \sigma_i^2}\end{aligned}$$

Such a simple closed form expression for the node update probability is not known for the ML estimator. But As we see in the last section, its performance is pretty close to the performance of the computationally much complex ML-estimator.

We can observe from (12) that the probability of a node taking either of the binary values is strictly positive for any given initial state. In other words, every transition in the t.p.m has a strictly positive probability and hence  $P > 0$  (element wise) in the presence of noise.

In a time variant Markov chain, one considers the average t.p.m. (denoted  $\bar{P}$ ) to characterize the consensus behavior, rather than one particular instantiation of the t.p.m. [2].

Clearly  $\bar{P} > 0$  (element wise), as  $P(t) > 0$  (element wise) for every channel instantiation under noise. This implies from Perron- Frobenius theorem that, the Markov chain has exactly one eigenvalue with absolute value equal to one and hence becomes irreducible and aperiodic with a unique stationary probability distribution. Hence, the steady state of the system is a *memoryless state*, meaning that the steady state doesn't depend upon the initial state in any manner. Hence, the system eventually forgets what its initial state is or what the initial majority is. In other words, we eventually reach an arbitrary

state decided by the structure of t.p.m. but not based on the majority of bits in the initial state.

Hence, the system is not expected to reach steady state quickly and should remain in the transient state as long as possible, at least as long as sufficient for the practical purposes. In such a case, the absolute value of the second eigen value of the t.p.m. ( $< 1$ , for a regular Markov chain) captures the notion of how long it takes to reach steady state and is expected to be close to one. As precise second eigen value computation is difficult in general, an approximation to the value would be desired for seeking insights into the system. It should be noted that as two eigen values of t.p.m. under zero noise are equal to one, as the noise variance approaches zero, second eigen value should asymptotically approach one. This would qualify an arbitrary eigen value to be at least a close approximation for the actual second-eigenvalue of the system. Though the second eigen value is not a complete metric to characterize the consensus performance, it will enable us in understanding of the behavior of the system in transient state.

### V. SECOND EIGENVALUE APPROXIMATION FOR THE AVERAGE TPM

*Property-1:* Let  $P$  be an arbitrary transition probability matrix of a Markov chain  $\{\mathcal{D}(t) \in \Phi : t \in \mathbb{Z}^+\}$ , where  $\Phi$  is an arbitrary discrete state-space. Let  $f$  be a scalar-valued function over the states in  $\Phi$  such that,

$$\begin{aligned}\mathbb{E}_\phi [f(\mathcal{D}(t+1) = \phi) | \mathcal{D}(t) = \phi^{(i)}] \\ &= \sum_{\phi^{(j)} \in \Phi} f(\phi^{(j)}) \Pr(\mathcal{D}(t+1) = \phi^{(j)} | \mathcal{D}(t) = \phi^{(i)}) \\ &= K f(\phi^{(i)}) + C_{\phi^{(j)}}, \quad \text{with } C_{\phi^{(j)}} = 0 \text{ if } K = 1, \quad (13) \\ &\quad \forall \phi^{(j)} \in \Phi.\end{aligned}$$

then  $K$  is an eigen value for  $P$  with the corresponding right eigen vector  $\underline{x}$  that satisfies  $P\underline{x} = K\underline{x}$  is given by:

$$\underline{x} = \begin{cases} \begin{bmatrix} f(\phi^{(1)}) \\ f(\phi^{(2)}) \\ \vdots \\ f(\phi^{(M)}) \end{bmatrix} + \frac{1}{K-1} \begin{bmatrix} C_{\phi^{(1)}} \\ C_{\phi^{(2)}} \\ \vdots \\ C_{\phi^{(M)}} \end{bmatrix}, & \text{if } K \neq 1, \\ \begin{bmatrix} f(\phi^{(1)}) \\ f(\phi^{(2)}) \\ \vdots \\ f(\phi^{(M)}) \end{bmatrix}, & \text{if } K = 1. \end{cases}$$

Property-1 can be verified as a special case of definition of an eigenvalue and the corresponding eigenvector to a stochastic matrix. Here, elements of the eigenvector are viewed as scalars generated by a function defined on the state space.

Now we try to derive the approximate expression for the second eigen value in a similar way of [2]. This uses a linear approximation to  $Q$  function and a naive choice of  $f$ .

*Theorem-1:* With i.i.d. noise with  $\sigma_i^2 = \sigma^2$  at each receiver, let the average dynamical system described by  $\bar{P}$  have eigen values  $\{\lambda_1, \lambda_2, \dots, \lambda_M\}$  with  $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_M|$ . Then the second eigen value  $\lambda_2$  may be approximated by:

$\lambda_2 \approx 1 - 2\bar{\gamma}_{\phi^{(1)}}^{(i)}$ , where  $\bar{\gamma}_{\phi^{(1)}}^{(i)} = \mathbb{E}_H [\gamma_{\phi^{(1)}}^{(i)}]$ , [refer (12)] which represents the mean probability of decision rule at node- $i$  choosing in favor of +1 when the system is in *all-zero* state  $\phi^{(1)}$ .

*Proof:* Let a scalar valued function  $f$  be defined on the state space  $\Phi$  as follows:

$$f(\phi) \triangleq \mathbf{1}^T \phi, \forall \phi \in \Phi.$$

which simply sums up all node's decisions which can be seen from the state vector  $\phi$  of elements  $[\phi_1, \phi_2, \dots, \phi_N]^T$  where each  $\phi_i = \pm 1$  is a decision taken by node- $i$ . Then, it is easy to see that

$$\mathbb{E}_\phi [f(\phi)] = \sum_{i=1}^N \mathbb{E}_\phi [\phi_i], \quad (14)$$

$$= \sum_{i=1}^N (\Pr\{\phi_i = +1\} - \Pr\{\phi_i = -1\}), \quad (15)$$

$$= \sum_{i=1}^N (2\Pr\{\phi_i = +1\} - 1) \quad (16)$$

Now, given an initial state  $\phi^{(i)}$ , by using the linear approximation  $Q(z) \approx 0.5 - z/\sqrt{2\pi}$  to (12), which is valid for sufficiently small values of  $z$  close to zero, we can write:

$$\begin{aligned} \gamma_{\phi^{(j)}}^{(i)} &\triangleq \Pr\{x_i(t+1) = +1 | \mathcal{D}(t) = \phi^{(j)}\} \\ &= Q\left(\frac{-1}{\sigma_i} \left\{ \frac{\sum_{j \neq i} |h_{ji}|^2 + \sigma_i^2}{\sum_{j \neq i} |h_{ji}|} x_i + \sum_{j \neq i} |h_{ji}| x_j \right\}\right) \\ &\approx \frac{1}{2} + \frac{1}{\sqrt{2\pi}\sigma_i} \left\{ \theta_{ii} x_i + \sum_{j \neq i} \theta_{ji} x_j \right\} \end{aligned} \quad (17)$$

$$\text{where } \theta_{ii} \triangleq \frac{\sum_{j \neq i} |h_{ji}|^2 + \sigma_i^2}{\sum_{j \neq i} |h_{ji}|}, \text{ and } \theta_{ji} \triangleq |h_{ji}|, j \neq i.$$

Assuming i.i.d. channels and  $\sigma_i^2 = \sigma^2$ , on averaging over channels gains from  $H$ ,

$$\mathbb{E}_H [\theta_{ii}] = \mathbb{E}_H \left[ \frac{\sum_{j \neq i} |h_{ji}|^2 + \sigma^2}{\sum_{j \neq i} |h_{ji}|} \right] \triangleq \kappa_1 \forall i \quad (18)$$

$$\text{and } \mathbb{E}_H [\theta_{ji}] = \mathbb{E}_H [|h_{ji}|] \triangleq \kappa_2, \forall j \neq i. \quad (19)$$

Correspondingly, the average probability becomes:

$$\bar{\gamma}_{\phi^{(j)}}^{(i)} \triangleq \mathbb{E}_H [\gamma_{\phi^{(j)}}^{(i)}] \quad (20)$$

$$\begin{aligned} &= \mathbb{E}_H [\Pr\{x_i(t+1) = +1 | \mathcal{D}(t) = \phi^{(j)}\}] \\ &\approx \frac{1}{2} + \frac{1}{\sqrt{2\pi}\sigma} \left\{ \kappa_1 x_i + \kappa_2 \sum_{j \neq i} x_j \right\} \end{aligned} \quad (21)$$

Now, conditioning over the current state of the system  $\{\mathcal{D}(t) = \phi^{(j)}\}$  and averaging over the mean transition probabilities in  $\bar{P}$ , equation (16) can be modified as,

$$\mathbb{E}[f(\phi) | \mathcal{D}(t) = \phi^{(j)}] = \sum_{i=1}^N (2\bar{\gamma}_{\phi^{(j)}}^{(i)} - 1), \quad (22)$$

$$= \frac{(N-1)\kappa_2 + \kappa_1}{\sqrt{2\pi}\sigma} \left( \sum_{j=1}^N x_j \right), \quad (23)$$

$$= (1 - 2\bar{\gamma}_{\phi^{(1)}}^{(i)}) f(\phi^{(j)}). \quad (24)$$

where,  $\phi^{(1)} = [-1, -1, \dots, -1]^T$ , the state of *all-zero*. Further, using property-1, we can conclude that  $(1 - 2\bar{\gamma}_{\phi^{(1)}}^{(i)})$  is an approximation to some eigen value of the system. From the exact equation (12) with  $x_i(t) = -1, \forall i \in \mathcal{N}$ ,  $\gamma_{\phi^{(1)}}^{(i)}$  tends to 1 as noise variance tends to zero. This concludes that it will serve as a close approximation for the second eigen value which shows the similar behavior.  $\square$

In the following section we present various simulation results that demonstrate the system's performance.

## VI. SIMULATIONS

We now see various simulation results that validate the system's working performance, also in comparison with the existing systems. For all the simulations, we choose an 8 node fully connected network.

We first compare the simple LMMSE algorithm performance with the ML algorithm to ensure that it is indeed a good alternative for high complexity ML algorithm. From figure-1, we can see that the LMMSE algorithm is a very good alternative to ML except at very low CNR's.

Next, We choose the performance metric of consensus algorithm as average probability of accurate-consensus (APAC), averaged over all possible channel states at each time instant. We then plot the APAC for the current DCP system and a system proposed in earlier papers in power-fair conditions. We choose an earlier scheme named Basic Affine Estimation (BAE) as baseline, to evaluate the performance of LMMSE algorithm, as shown in fig-2. The BAE algorithm uses a round-robin broadcasting scheme of data transmission for an iteration. This system is thoroughly analyzed and is shown to achieve a good level of performance in consensus [2]. It can be seen from figure-2 that DCP is shows a significantly better and consistent performance than the BAE algorithm at low SNRs.

We then finally simulate the average t.p.m.'s at various CNR's using Monte-Carlo simulations and calculate precise second eigenvalues, so that we can compare and validate the approximate second eigenvalue expression derived in earlier section-V. Accordingly, figure 3 shows that the approximation matches well with the actual eigenvalues obtained by Monte-Carlo simulations.

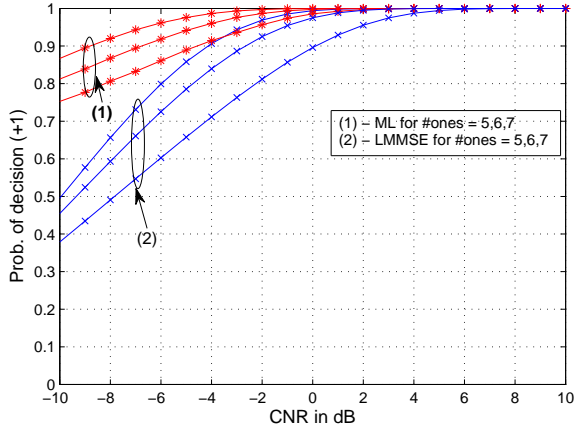


Fig. 1. Comparing the performance of LMMSE algorithm with ML for different initial majorities in a network with 8 nodes

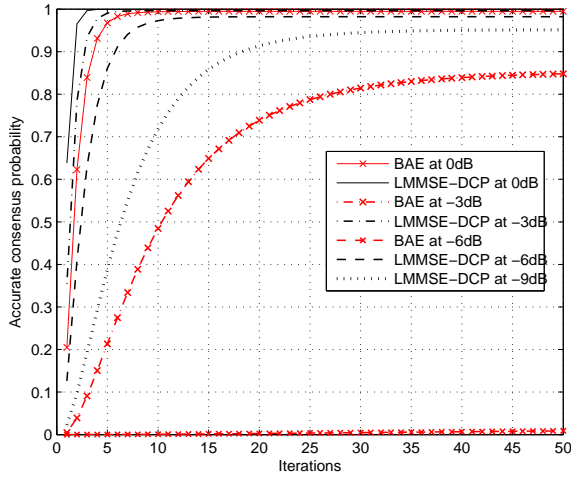


Fig. 2. Comparing the performance of DCP LMMSE algorithm with BAE algorithm, when 6 of 8 nodes are voting +1 initially

## VII. CONCLUSION

We presented a new model for detection consensus based on distributed consensus, that performs well compared to the earlier schemes, especially at low CNR's. We derived a simple linear decision rule based on LMMSE estimation, that matches with ML rule quite well at reasonable CNR values. We shown that the system of consensus eventually reaches a memoryless state by staying in transient state as long as depicted by the closeness of second eigenvalue to one. We derived an approximation to the second eigenvalue to seek insights into the performance of algorithm. We see that this approximation matches quite well with the actual value.

## REFERENCES

[1] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proc. of American Control Conference*, vol. 3, June 2005, pp. 1859–1864.

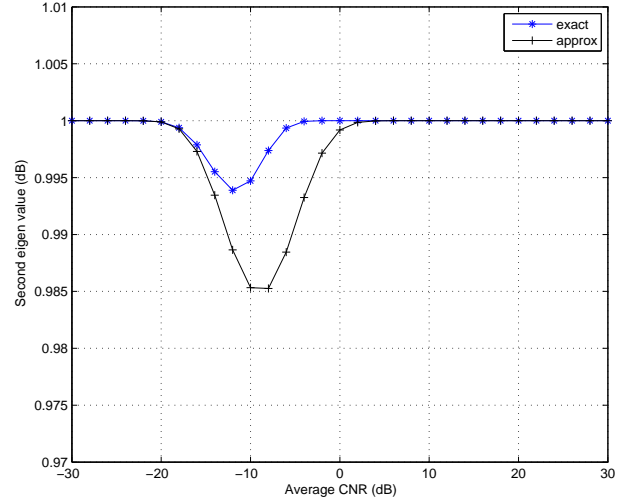


Fig. 3. Comparing the approximation of second eigen value with actual value at various CNR's in a network with 8 nodes

- [2] Y. Mostofi and M. Malmirchegini, "Binary consensus over fading channels," *IEEE Trans. Signal Processing*, vol. 58, no. 12, pp. 6340–6354, Dec. 2010.
- [3] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: design, analysis and applications," in *INFOCOM 2005*, vol. 3, Mar. 2005, pp. 1653–1664.
- [4] —, "Randomized gossip algorithms," *IEEE Trans. Information Theory*, vol. 52, no. 6, pp. 2508–2530, June 2006.
- [5] T. Aysal, M. Yildiz, and A. Scaglione, "Broadcast gossip algorithms," in *IEEE Information Theory Workshop (ITW'08)*, May 2008, pp. 343–347.
- [6] D. Shah, "Network gossip algorithms," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'09)*, Apr. 2009, pp. 3673–3676.
- [7] C. Steffens and M. Pesavento, "A physical layer average consensus algorithm for wireless sensor networks," in *International ITG Workshop on Smart Antennas*, Mar. 2012, pp. 70–77.
- [8] F. Benezit, P. Thiran, and M. Vetterli, "Interval consensus: From quantized gossip to voting," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'09)*, Apr. 2009, pp. 3661–3664.
- [9] —, "The distributed multiple voting problem," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 791–804, Aug. 2011.
- [10] Y. Mostofi, "Binary consensus with gaussian communication noise: A probabilistic approach," in *46th IEEE Conf. on Decision and Control (CDC'07)*, Dec. 2007, pp. 2528–2533.
- [11] K. V. K. Chaythanya, R. Annavajjala, and C. R. Murthy, "Comparative analysis of pilot-assisted distributed co-phasing approaches in wireless sensor networks," *IEEE Trans. Signal Processing*, vol. 59, no. 8, pp. 3722–3737, Aug. 2011.
- [12] R. Annavajjala and C. R. Murthy, "Pilot-assisted distributed co-phasing for wireless sensor networks," in *6th Annual IEEE Comm. Soc. Conf. on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'09)*, June 2009, pp. 1–8.
- [13] R. Mudumbai, G. Barriac, and U. Madhow, "On the feasibility of distributed beamforming in wireless networks," *IEEE Trans. Wireless Communications*, vol. 6, no. 5, pp. 1754–1763, May 2007.